BCIS 4610 FINAL REPORT

By : Team Logical Partition LLC.

Hugh Hunton, Cody Morris, James Konderla, Mark Wallace

AIDAN GRAY

Final Report Content

Overview Page 1

Executive Summary Page 2

Goals of The Project Page 3

Project Scope Page 4

Future System Recommendations Page 5

System Requirements Page 6

Functional Requirements of the New System Page 7

Data Requirements Page 9

Non-Functional Requirements Page 10

System Design Specifications Page 11

System Acquisition Strategy Page 12

Architectural Design Page 13

Database Design Page 14

User Interface Design Page 16

Interface Design Standards Page 17

User Interface Design Standards Page 18

Program Design Page 19

Implementation Issues Page 20

Prototype Page 21

OVERVIEW

Executive Summary

With the assistance from Aidan Gray employees, Team Logical Partition has successfully gathered user requirements, developed prototypes, conducted testing and implemented a solution which we know will drastically reduce operational costs.

Through a series of interviews, questionnaires and observation, Logical Partition recognized redundancy in the data entry process. As a team, we made it our goal to streamline this process by eliminating redundancy with the ultimate goal of increasing user productivity while reducing operational costs. Our primary design objective was to simplify everything by removing distractions and redundancies from the user interface.

We were very excited by with the feedback we received from our first prototype. Although it wasn't used in a production environment, our initial testing suggested a reduction in time spent on data entry by 66%. The users were extremely happy with this prototype and we knew we were going in the right direction. After receiving such a positive response to our initial testing, we turned our attention to implementing our software in the most efficient way possible.

Several weeks later we have successfully deployed the software on Aidan Gray's network. This deployment was done without the purchase of any commercial software, or changes to the hardware configuration. That means there were no additional costs incurred, which also means there will not be any increase in maintenance costs. Training was conducted over the course of one business day, without any significant interruption in day-to-day activities.

Overall the results of this project have been extremely positive. We know that we have provided Aidan Gray with the best solution to their unique problem. We are happy to highlight a few of the benefits that Aidan Gray will enjoy:

- Reduced data entry time
- Increased data accuracy
- Reduced operational costs
- Increased employee satisfaction
- Streamlined data entry process

- **All accomplished at 1/10th the cost of the next alternative**

.

The goal of the project

Logical Partition has identified the inadequacy in Aidan Gray's existing business process. The team's goal is to address this issue by reengineering existing business processes to eliminate redundancies and increase efficiency.

Rather than attempt to deploy another large costly information system, Logical Partition has decided to design a middle-ware utility. The goal of this utility is to eliminate the requirement of updating three databases. If the team is successful in this goal, Aidan Gray will only need to enter product updates into the single middle-ware utility.

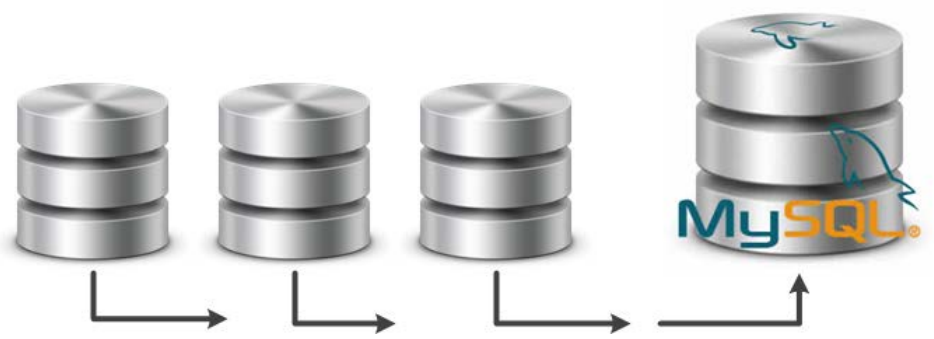
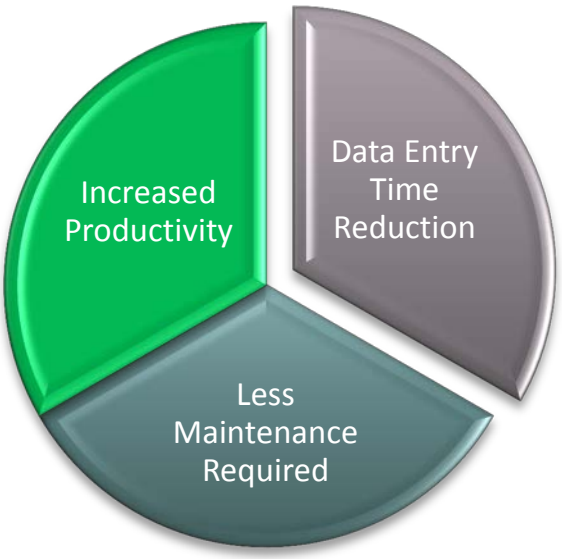


Figure 1: Goals of the project

Benefits for the client/other beneficiaries

Estimations suggest that the implementation of this application will create a 66% reduction in time required to update all three databases. The benefits to the business owners will be a significant increase in productivity. End-users will have more time to focus on other duties, creating greater satisfaction and efficiency. Customers, clients, and inventory workers will enjoy greater accuracy in product information due to the elimination of potential data entry errors. Additionally, the cost and time required to design and implement this application is only 10% of the development costs of alternative solutions. All users will enjoy this solution much sooner than the alternatives, and at a fraction of the cost. There is also no increase to maintenance or hardware requirements, which means efficiency is increased while cost of operation remains the same.



Scope of the project

Logical Partition's objective in this project is to create a middle-ware application between the three existing databases Aidan Gray currently uses. These three databases will be referred to as "in-house", "excel", and "web". Any database added to Aidan Gray's infrastructure in addition to the three mentioned above is beyond the scope of this project. Additionally, the application will be deployed on existing hardware within Aidan Gray's office. The application is not guaranteed to work if there is significant modification to the hardware, or the network configuration on which it operates. Reconfiguring the application is only possible through rewriting source code, which is not available to the end user. Creating a user-configuration interface is also beyond the scope of this project.

The implementation of Logical Partition's middle-ware solution is intended to reduce redundancy in data entry. Specifically, the goal is to provide a single interface in which a user can create new entries. After the user populates the required fields, it will be pushed live across all three databases. It is important to note – the application will only allow users to create new entries, or alter entries previously created using the application. The constraints of the project do not allow for synchronization of any data located in any of the three databases prior to implementation. More specifically this means any old records will need to be edited the old way.

The application will be written to work entirely through a web browser. No new software or hardware will be required. Users who currently have access to edit the three databases will use the same login information to gain access to the application. There will be a back-end feature in which an administrator will be able to create, edit or delete user login credentials. This administrator interface will also be entirely web browser based.

Key deliverables

- System Specification
- Hardware & Software Specifications
- Interface Design
- Physical Process Model
- Database & File Specification
- Physical Data Model
- Programs & Documentation
- Migration Plan:
- Support Plan
- Project Proposal
- Interim project Report
- Final Project Report

Limitations

- No Synchronization of *existing* data, only synchronization of new data will occur in this system.
- Client will NOT be able to edit the form and/or interface themselves, further editing of the pre-designed system will be done through consulting.
- No structural changes to the database will be done after the Design Phase.
- No changes will be allowed to database hardware to prevent unforeseen program or database errors.
- Administrative access should be limited to keep from accidental deletion or breakage in the system. User accounts and passwords will be issued as needed.
- No Remote Connection to the Database will be allowed due to security reasons.
- After Implementation, the prior system will be seen as a separate entity and changes to the prior system will not affect the current system.

Future System Recomendations

Team Logical Partition, while implementing this project, discovered several areas where Aidan Gray might like to improve in the future. We believe that these areas will drastically increase system efficiency, remove any possibility of extended downtimes due to system failure, and overall increase productivity and profits across the board.

System Upgrades

The first major area for possible future system improvements, that will aide in company-wide productivity, is the upgrade of the various computer systems that Aidan Gray uses. We believe that by upgrading both the client and server hardware and software to the latest systems would improve the lifetime of Aidan Gray's systems. We recommend this once every 3 to 5 years in order to keep Aidan Gray from expending monetary resources while enabling Aidan Gray to take advantage of the latest technologies.

Database Consolidation

While the current system we are implementing is a drastic increase in productivity and efficiency, it is a temporary fix only, as we have stressed throughout the development of this system. Our recommendation is a complete database overhaul and consolidation which, we feel, would provide an even greater increase in productivity, efficiency, and profits than will be witnessed by the implementation of this project itself.

System Redundancy and Business Continuity

Every business has essential functions that must continue on a daily basis to keep the business flowing and Aidan Gray is no different. We recommend a Business Continuity Planning overhaul, starting with the implementation of a System Redundancy plan that should include backup facilities off site to minimize downtime and loss of profits in a catastrophic situation such as fire, floods, or any other number of natural disasters.

We have many more recommendations for Aidan Gray that would increase the productivity and profits of Aidan Gray but we feel that these three are the most important and should be addressed by Aidan Gray within the next few years in order to keep Aidan Gray's ability to service it's customers at it's highest levels as the business grows and expands beyond it's current needs.

SYSTEM REQUIREMENTS

Goal of the system, high level description

The purpose of the system is to streamline the data entry process by combining several DBMS interfaces into a single interface. This single interface, or form, will allow users to populate the fields with appropriate data and submit it to the three different databases (in-house, web, excel). The new system will differentiate the information, and only push the information to the appropriate destination. For example, the user enters a description, dimensions and price for a table. The description may only be sent to the web, the dimensions may go to excel, and the price may go to the in-house database. The new system will also allow users to query based on SKU (stock-keeping-unit), and have all information from the three databases consolidated into a single interface. From this interface, users will be able to edit or delete information, and send the changes back to the applicable databases.

Description of system functionality

Product Query

Users will be able to query product from the various databases by using product SKU. Information from all three databases will be displayed in a single interface.

Edit Product

Users will be able to edit existing data and save the changes.

Resolve Discrepancies

Alerts users when there is a variance between databases on identical product characteristic. For example query for SKU 12345 returns table 'A' with height 'X' from one database, and returns table 'A' with height 'Y' from another. The alert serves as a reality check, and allows the users to select which value is the true value.

New Product

Users can create new products.

Provide Data to Appropriate Database

Discriminates the information based on its field and sends it to the appropriate database.

Relationship to other systems

The new system will work with existing hardware, software and databases. There will be no modification to any of the existing infrastructure.

The other systems that will be a part of the new system's process are:

In-house database and its current structure

- MySQL

Excel database

- .xls files

Web database

- MySQL

Each of the above systems will be accessed through the new system and will give the user read/write/edit privileges.

Brief description of the key types of data that is being created , stored and manipulated by the system

In general, our data consists of product information and Quantity-On-Hand counts. This information, in its current state, is spread across 3 databases, which we will describe below:

In-House Database: This database is local and is based in Microsoft Access 2003. This database holds all information regarding the products. This database, while Access based, is accessed through a shared folder on the network and is queried using basic SQL queries. This database is known as A on our ERD (see appendix – UnJoined ERD).

Web-Database: This database is used for the AidenGrayHome.com website. This database is stored off-site, on the web-provider's server, and most of the data in this database is a duplication of the in-house database. While it's data is majorly a duplication of the local database, the Web Database has no connectivity to the in-house system and must be updated manually. This database, according to the publisher, is MySQL based and is supplied to the web in HTML and Javascript format. This database is referenced as B in our ERD (see appendix – UnJoined ERD).

Excel Database: This database is used exclusively by the sales representatives and is housed in a Microsoft Excel 2003-based file that is uploaded to a third party source (www.repzio.com) and supplies data to the iPad-based sales application. This database also has no connectivity to the in-house system and, for all intents and purposes, is considered its own entity. This database, according to Repzio, is supplied to the user in a Javascript and HTML format that draws on the Excel file after it's conversion to a MySQL database (this is outside the scope of our project but is important enough to Aiden Gray Home that it should be noted) This database is referenced as C in our ERD (see appendix – UnJoined ERD).

As part of our solution to Aiden Gray Home's problem of database integration we have decided to focus the In-House database as the main system. As such, according to our JOINED ERD, we have decided to join the databases on their similar fields and, as part of the same display, query the different, or system-specific, fields of each database to provide the user form.

Entities

The main entities involved in this system, according to our ERD (see appendix – Joined ERD), are outlined below (for descriptions, see appendix – Joined ERD):

Item Info Table – This table stores all the Item Information and consolidates the fields from the 3 separate databases (listed on the previous slide). It's attributes are listed below:

- ItemInfo(**SKU (ABC)**, Item_Name (ABC), ***Department_ID (AB)***, Item_Weight (AB), Dimensions (ABC), Materials (AB), Color (AB), Category (C), Image_URL (C), Discontinued (C), Min_Order_Q (C))

Department Table – This table stores the information about each different department and pulls this from each database where it is listed, consolidating it. The attributes are listed below:

- Department(**Department_ID (AB)**, Department_Name (AB), Department_Description)

Pricing Table – This table stores the pricing, as well as minimum purchase requirements, of each separate item listed in the system. Part of it's primary key, the SKU, references the itemInfo table while the rest of the attributes contain the data pulled from the different tables and consolidate it into this table. The attributes are listed below:

- Pricing(***SKU (ABC)***, **Type (A)**, Retail_Cost (BC), Wholesale Cost (BC), Min_Req_Pur (BC), Cost (A))

QOH Table – This table lists the current Quantity on Hand, as well as the re-order information, for each item in the system. The SKU portion of the primary key references the ItemInfo table while the Loc_ID references the physical location of the item in the warehouse. It's attributes are as follows:

- QOH(***SKU (ABC)***, **Loc_ID (AB)**, Re_Order_Point (AB), Num_On_Hand (AB))

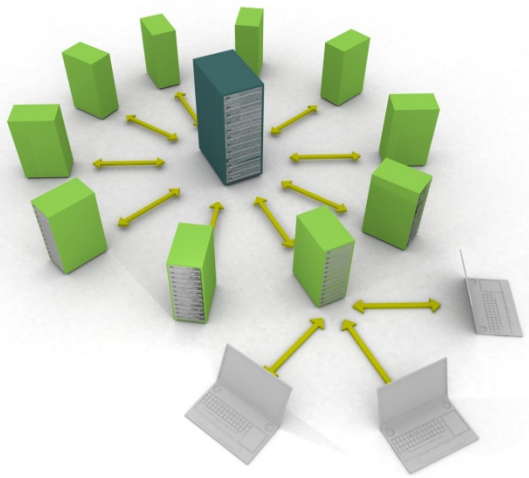
Relationships

For the Database, according to the ERD (see appendix – Joined ERD), consist of three major Relationships:

Item's Department – This relationship is between the ItemInfo and Department tables and is used as a Foreign Key in the ItemInfo table, to show which department an Item Belongs to.

Item's SKU – This relationship is between the Pricing and ItemInfo Tables and is used, as a Foreign Key and portion of a concatenated Primary Key in the Pricing table, to generate pricing information, such as wholesale and retail prices, for each separate Item based on the Item's SKU.

Item's SKU 2 – This relationship is between the ItemInfo and QOH tables and is used, as a Foreign Key and portion of a concatenated Primary Key in the QOH table, to generate Quantity On Hand information for each item Based on it's SKU.



Key Considerations/Architecture

While interviewing with the management of Aiden Gray Home Inc. it was determined that the best option, based on environmental factors and user needs, would be a Server-Based system. During the interview and data-gathering sessions we were also able to determine that Security, while important, is not the most pressing issue. Instead, the company must *always* have access to their data for continued and uninterrupted operations. Because of this we have decided that the server will be accessible locally without remote control but will also be connected to the internet to simplify updating of the web database. We also determined that a basic user interface will be sufficient when accessing data. In the following sections we will outline the key portions of the Non-Functional system requirements.

User Interface Requirements

According to our interview sessions, the User Interface for our system should concern itself with Usability rather than aesthetics. As such, we have designed a basic form layout on an HTML-based user interface to meet the basic requirements of this project. We have also decided, based on the relative ease of implementation, to base the interface off of the corporate colors of Orange, white, and Slate.

Mobile Integration, while originally an issue, is made simpler due to an integration package provided by Repzio that allows our interface to automatically validate our user requirements and update the web database, stored on Repzio servers, from the centralized user interface. Since speed and availability is an issue, we have decided to base our User Interface program on the server and to program it using basic HTML and Javascript tags without “bells and whistles”.

Performance

During our interview sessions we determined that speed is one of the most important issues in regards to the database system. The transaction volume, after reviewing sales data from the last year, seemed relatively standard so we have decided to implement a basic server-based system. We have also determined that the current database, while relatively small, must be left room to expand as Aiden Gray Home Inc. grows and, as stated in the key considerations, the system must *always* be available. To solve these issues we have determined that the server should be based on-site rather than remotely.

Security

According to our various interviews, Aiden Gray’s current database systems have built-in user validation with same-password validation per user across the 3 systems. As such, the login system will be simplified in our User Interface by validating across all 3 servers at once without the need to re-enter login data for each. We have also determined that, due to the necessity of uptime for the server, we will implement an Antivirus system on the system to prevent issues from outside sources. To prevent internal Security Issues, we have also determined that the Server will be stored in a locked server room.

Other Requirements

Based on the fact that Aiden Gray Home Inc. is based in Texas and deals internationally, we have decided to implement basic functionality for French and Spanish. According to the complexity of each language, though, we have decided that the interface itself will include this functionality, while the database will still be English-Based instead of being forced to translate each specific product to a different language. We have also determined that, based on the interviewers’ response of ease-of-use, that we will not allow customizability of the system or user interface after implementation.

SYSTEM DESIGN SPECIFICATIONS

Review of options

Due to the specific nature of the software being developed, there is no commercially available alternative. Modification of any other commercial software would be quite extensive and would not be financially justifiable. Custom development remains the fastest and least expensive option available and it ensures that Aidan Gray will receive a product built specifically for their needs.

In the long run, the ideal solution would a consolidation of all databases used by Aidan Gray. However, due to the financial burden of such and undertaking, and the time constraints in place, the option of consolidation is not ideal. Consolidation of the databases would also require new user interfaces and extensive testing, which significantly disrupts business operations. Given these constraints, the best solution for Aidan Gray is the custom development of a middleware software.

The proposed solution is the custom development of a middleware application that will make use of all legacy hardware and software systems, while still providing the functionality necessary to improving Aidan Gray's inventory management business processes. The key benefits to this solution are:

•Due to the light weight nature of the software, purchasing hardware upgrades will not be necessary. The software will run entirely on legacy systems, which further reduces development costs and future maintenance requirements.

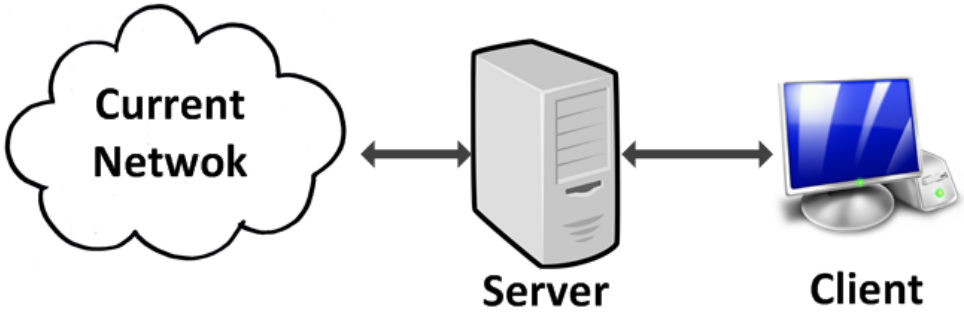
•No new software packages or licenses are necessary for development.

Comparison of options

Option	Pros	Cons	Recommendation
Database Consolidation	Improved data integrity	Cost, development time, new hardware and software requirements, significant testing required	Ideal for long run use, but poor choice with budget and time constraints
Commercial Software Modification		Cost, development time, dealing with vendors	Do not recommend
Middleware Application	Inexpensive, short development time, meets all specific requirements	Not versatile. Only intended as a hold over until database consolidation can be completed.	Best solution for short term use with given time and cost constraints

Architecture design

The recommended architecture type consists of an in house client-server based model. The main server will consist of a basic, Windows Server 2003 Active-Directory configured system that will join with the current corporate domain without incident. This server will host The Apache and MySQL servers and will allow for future growth and increased efficiency within the company without upsetting the current infrastructure, a main requirement from the current management. Along with the server, since the current corporate environment does not support many clients, we recommend the purchase of a basic Windows 7-based PC for interfacing with the server while maintaining data security and integrity. This server would need Mozilla Firefox and MySQL command-line client for database administration purposes while maintaining the central database on the secured server.



SW and HW recommendations

	Client	Server 1	Server Upgrade
Operating System	Windows 7 Professional	Windows Server 2003 Standard	Windows Server 2008 Standard
Specialized SW	Mozilla Firefox, MySQL Client	Apache Server, MySQL Server	Apache Server, MySQL Server
HW	Dual-Core Intel-based PC	Dual-Core Intel-based PC	Quad-Core Intel-Based Server

DB design consideration

As you might notice in the Appendix, our System Database is a compilation of several actual databases. It functions as a temporary storage, pulling data from all the separate user databases, outlined below, and consolidating them for the user's session. The data is pulled based on SKU that the user queries and, upon termination of the session, is deleted. On the following pages we have outlined both the database design through both physical and conceptual views.

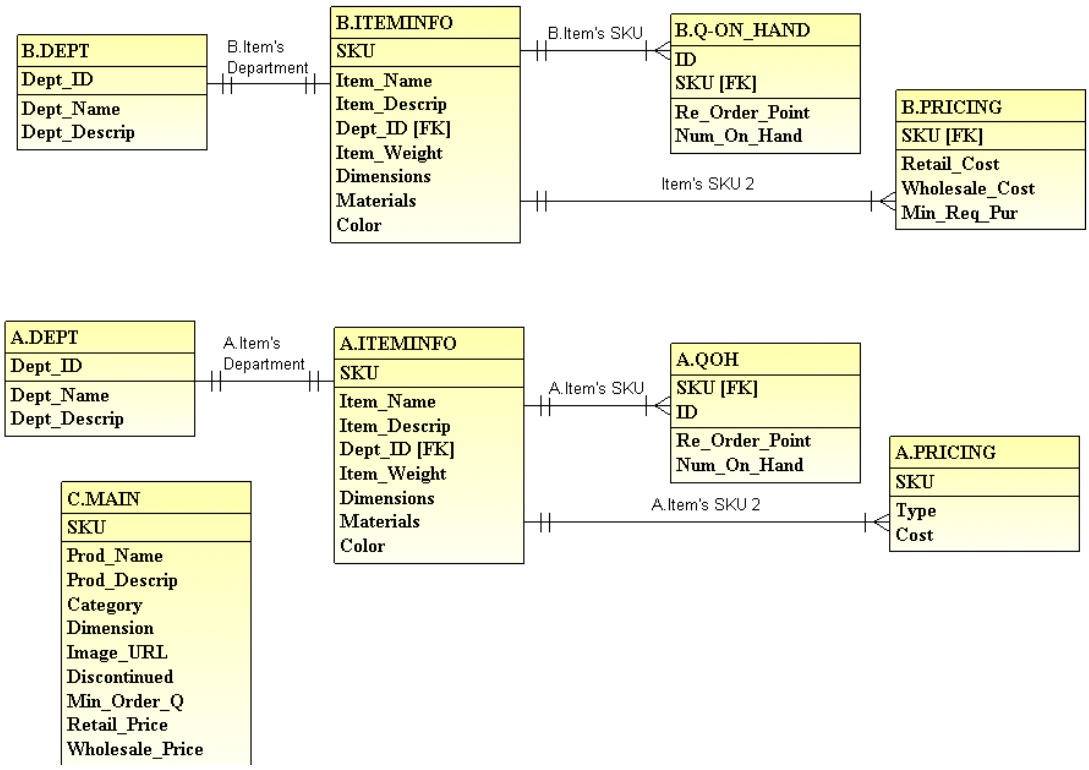
Data Store	Physical Storage
D1 – In House (A)	MySQL Database (On-Site Windows Server 2003-Based machine)
D2 – Web Database (B)	MySQL Database (Stored on a third-party web server at AidanGrayHome.com)
D3 – Excel (C)	Excel Spreadsheet (Stored locally, uploaded to a third party at Repzio.com)

As you can see from the table above, Aidan Gray's current database system is divided into 3 separate entities, an In-House database in MySQL, a Web, MySQL Database that is stored by a third party for Aidan Gray's website, and an Excel Spreadsheet that is stored locally and uploaded to a third party source, Repzio.com, for use in the Aidan Gray mobile app. We have outlined these databases conceptually in an Entity-Relationship Diagram (ERD) on the following page (see top figure).

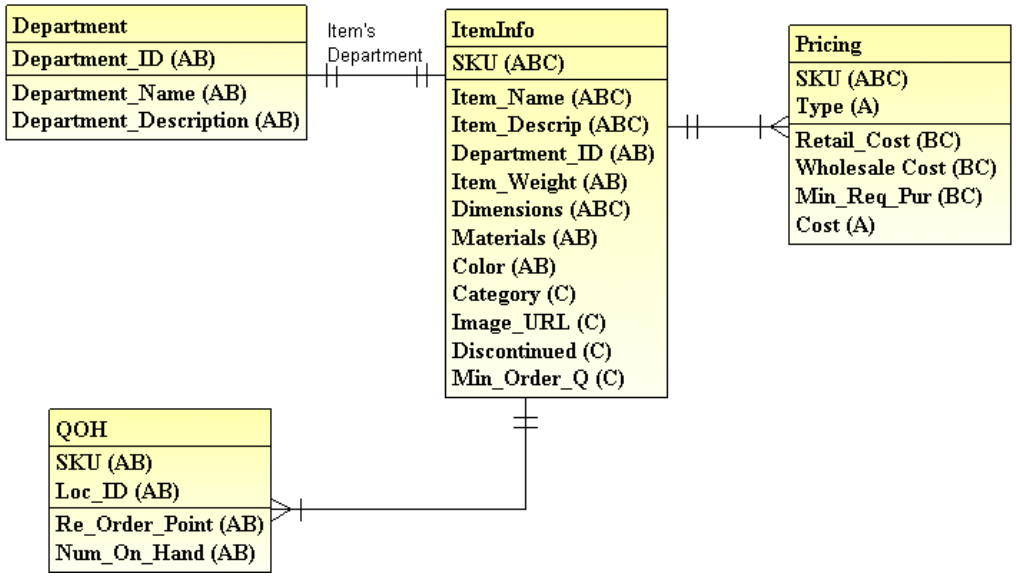
Our system is designed to take these various databases and, through user queries, pull them all into a temporary storage for editing, viewing, and updating by Aidan Gray's workers and associates. To do this we created a temporary storage database that combines common data based on entries in the In-House database (upon implementation all In-House and Web Database discrepancies will be solved) and non-common attributes from all databases based on their respective sources. The final product is laid out in a conceptual Entity-Relationship Diagram (ERD) on the following page (see bottom figure). This temporary data storage system will be implemented using MySQL on the company's current Windows Server 2003-based machine as a separate Scheme than the current In-House database to prevent unwanted editing or changes to the current system, as specified in our Project Limitations on the scope page.

Database Conceptual Designs (ERD)

Aidan Gray - Actual Database



Aidan Gray - System (Temporary Storage) Database



Key assumptions and principles

The interface design will reflect the goal of the software itself; streamline data manipulation and minimize user effort. In keeping with this theme, required user knowledge will be minimal. Users of the current system already possess the required knowledge of forms and data entry, so training requirements will be minimal.

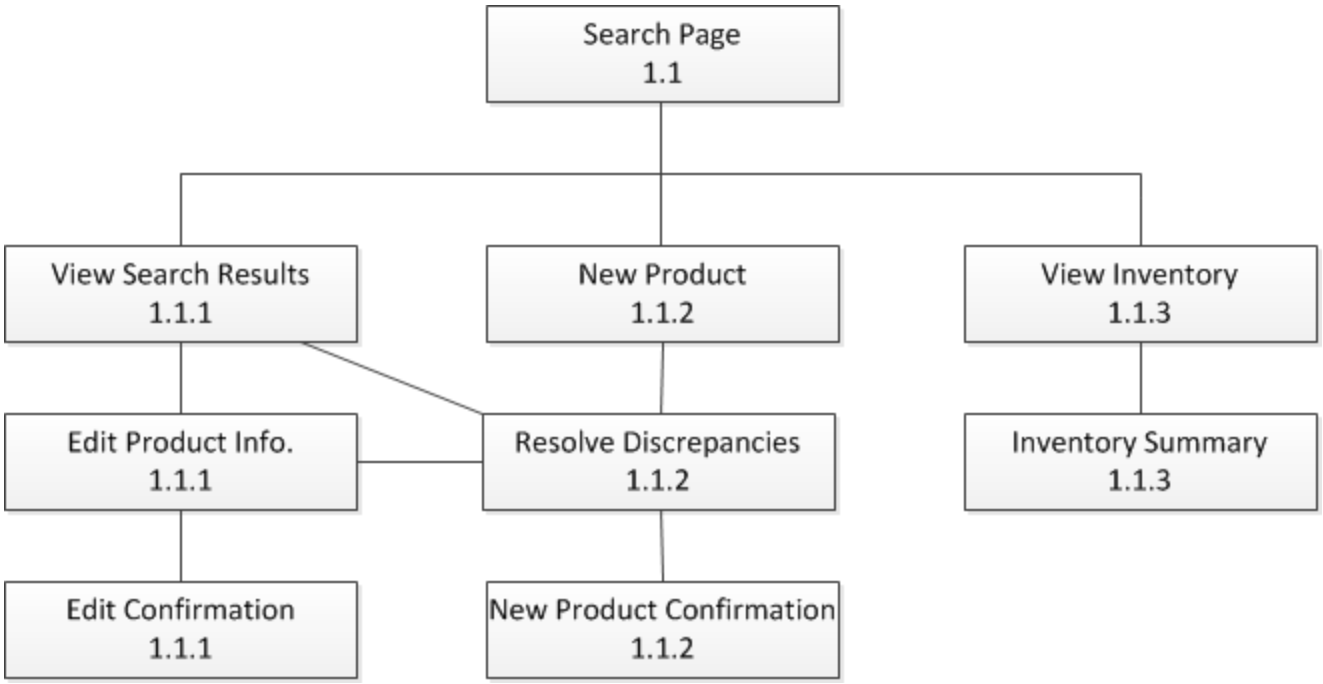
Hardware input and output device requirements include keyboard and monitor. To increase input speed, the forms are designed to be completely editable with just a keyboard. A mouse is not necessary, but may make the transition easier for some users.

The new software is not dependent on any particular operating environment or operating system. Any OS with an internet browser will be sufficient to run the new software.

Key principles and considerations in interface design:

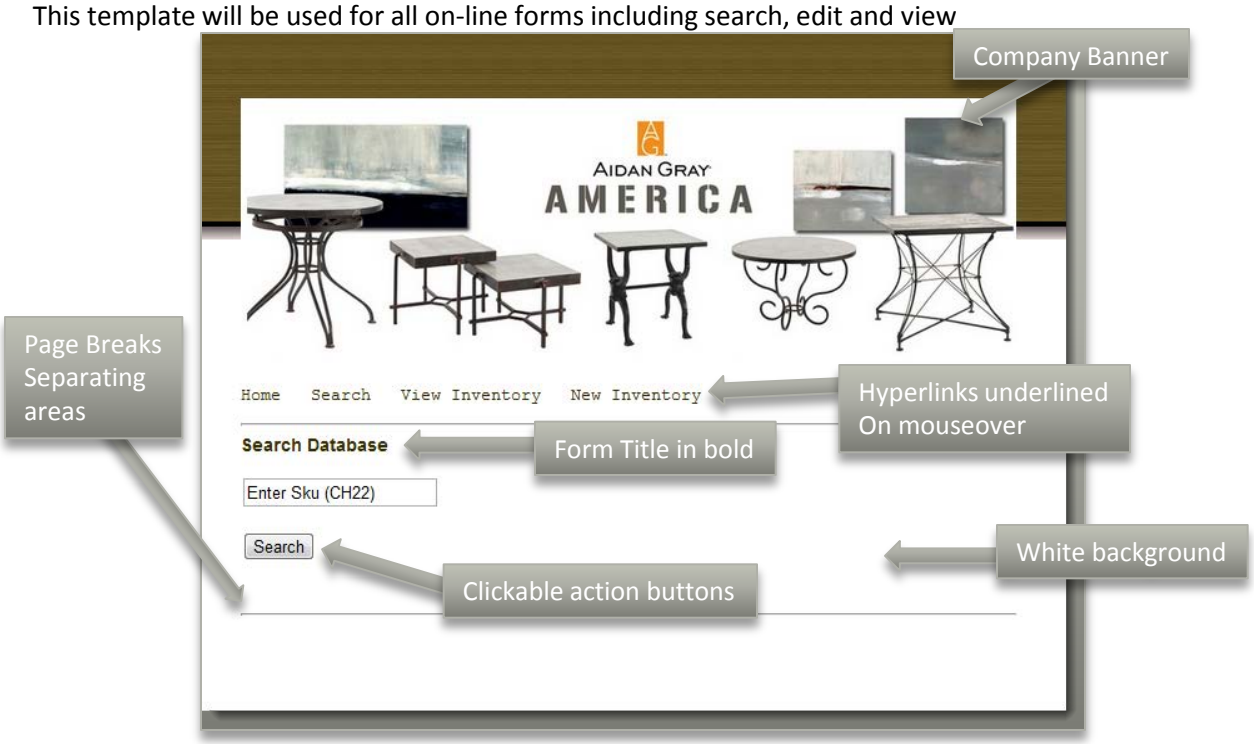
- Minimal graphical displays
- Minimal user effort
- Minimal learning curve
- Minimal resource requirements
- Maximum speed of input
- Maximum data accuracy
- Zero interface latency
- Zero data entry redundancy

Navigation structure chart



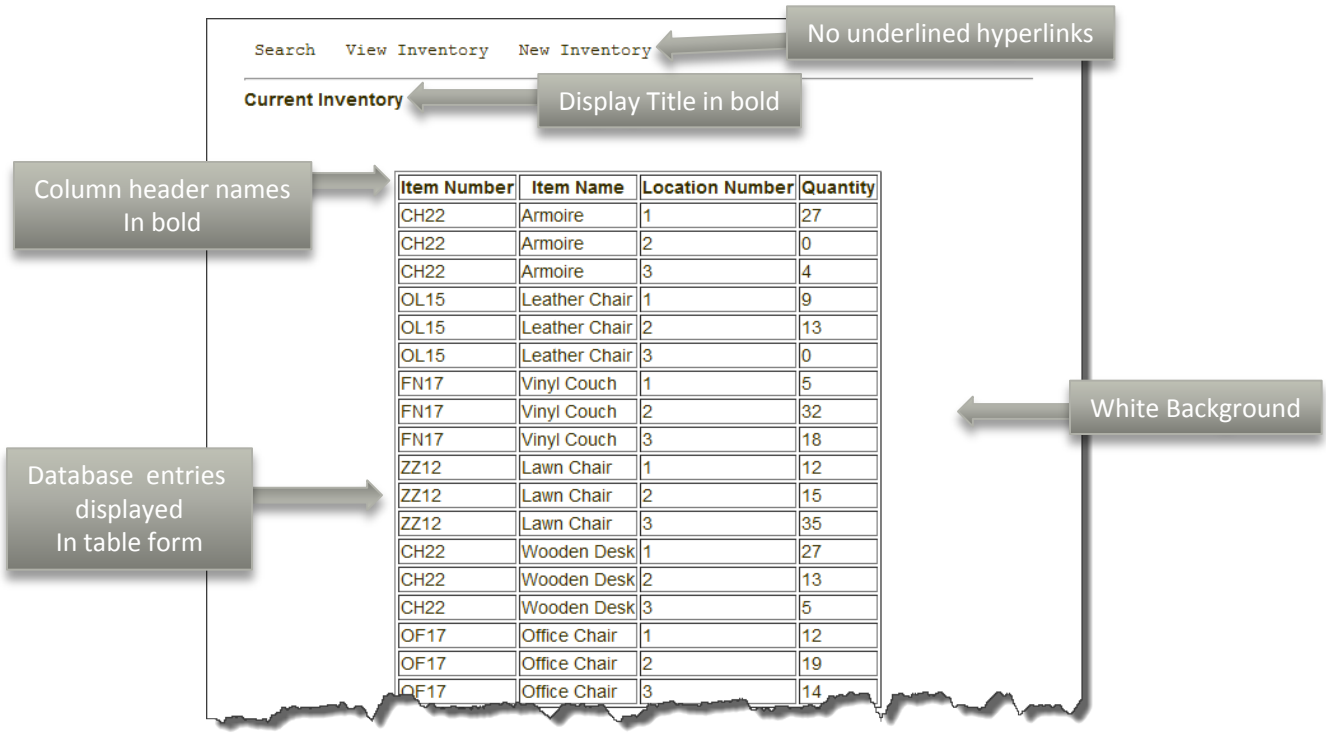
Standard Form

This template will be used for all on-line forms including search, edit and view



Inventory View

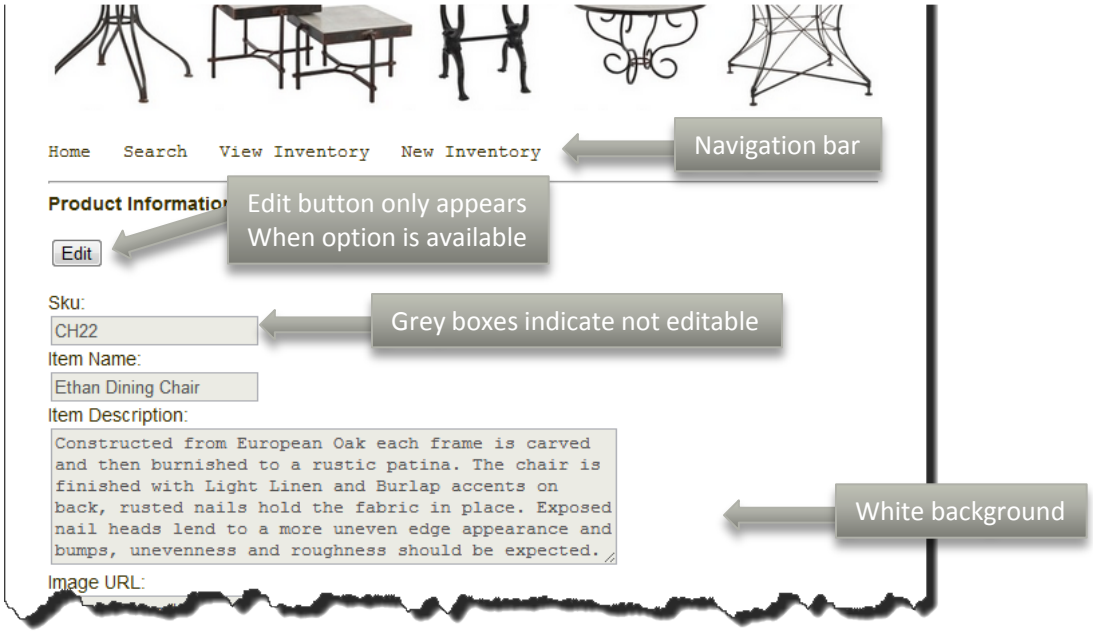
Standard layout for displaying product quantity information in databases



Item Number	Item Name	Location Number	Quantity
CH22	Armoire	1	27
CH22	Armoire	2	0
CH22	Armoire	3	4
OL15	Leather Chair	1	9
OL15	Leather Chair	2	13
OL15	Leather Chair	3	0
FN17	Vinyl Couch	1	5
FN17	Vinyl Couch	2	32
FN17	Vinyl Couch	3	18
ZZ12	Lawn Chair	1	12
ZZ12	Lawn Chair	2	15
ZZ12	Lawn Chair	3	35
CH22	Wooden Desk	1	27
CH22	Wooden Desk	2	13
CH22	Wooden Desk	3	5
OF17	Office Chair	1	12
OF17	Office Chair	2	19
OF17	Office Chair	3	14

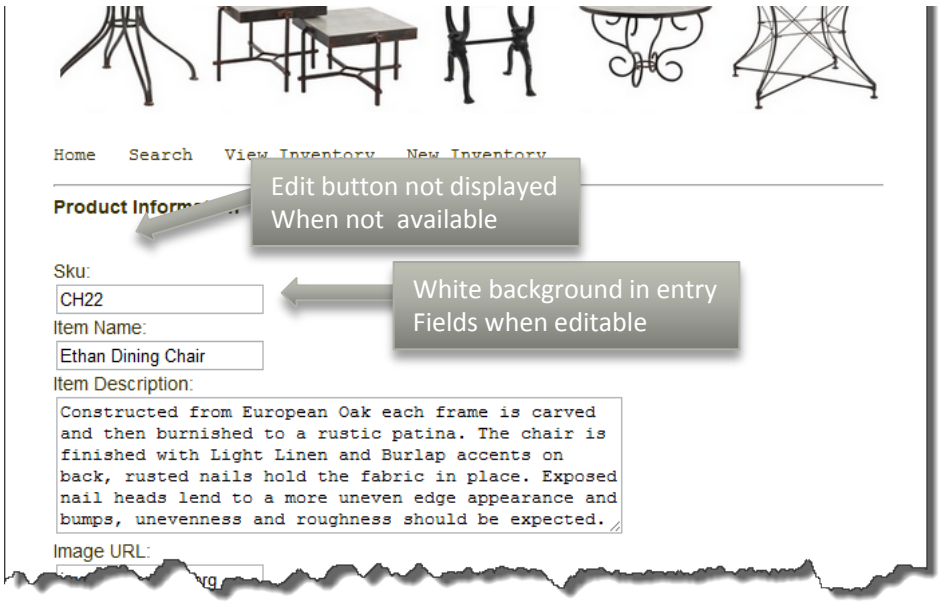
Standard Form Style

This is the standard view of a non-editable form


Home Search View Inventory New Inventory
Navigation bar
Product Information
Edit button only appears
When option is available
Edit
Sku:
CH22
Grey boxes indicate not editable
Item Name:
Ethan Dining Chair
Item Description:
Constructed from European Oak each frame is carved and then burnished to a rustic patina. The chair is finished with Light Linen and Burlap accents on back, rusted nails hold the fabric in place. Exposed nail heads lend to a more uneven edge appearance and bumps, unevenness and roughness should be expected.
White background
Image URL:

Form Interaction Behaviors

Standard view of a form once editing has been enabled


Home Search View Inventory New Inventory
Edit button not displayed
When not available
Sku:
CH22
White background in entry
Fields when editable
Item Name:
Ethan Dining Chair
Item Description:
Constructed from European Oak each frame is carved and then burnished to a rustic patina. The chair is finished with Light Linen and Burlap accents on back, rusted nails hold the fabric in place. Exposed nail heads lend to a more uneven edge appearance and bumps, unevenness and roughness should be expected.
Image URL:

Considerations in Program Design

Key consideration regarding program design

Programming language used to implement different processes in the system

	Level-0 Process		Level 1 Process	Programming language, other implementation technology
1.	Product Query	1.1	1.1 Create new product information	HTML Java script, JSP
		1.2.	1.2 Edit existing product information	HTML Java script, JSP
		1.3	1.3 Resolve discrepancy	HTML Java script, JSP
		1.4	1.4 Update databases with new information	HTML Java script, JSP
2.	New Product	2.1	2.1 Create New Product info	HTML Java script, JSP
		2.2	2.2 Provide new Info to database	HTML Java script, JSP
3.	Edit Product	3.1	3.1 Edit product information	HTML Java script, JSP
		3.2	3.2 Update edited product info to database	HTML Java script, JSP
4.	Provide data to separate Databases	4.1	4.1 Takes new data and edited data and updates each individual database and receives a confirmation when complete.	HTML Java script, JSP
5.	Resolve Discrepancy	5.1	5.1 Notify/correct discrepancy	HTML Java script, JSP
		5.2	5.2 Check and edit information with each database and receive the response	HTML Java script, JSP

Testing plan

We decided to test editing existing product information and creating a new product with new information. The tester was Hugh Hunton and he conducted the tests on April 15, 2012. Hugh Hunton also created the test scenarios and executed the testing. One issue we were able to see being a problem is that if you lose network connectivity while entering in product information you will lose all of the data you added and will have to start over again. Another would be that our system needs to be able to integrate with the login page and the current existing databases.

Documentation

The documentation we will be using is a user manual. The user manual will explain to the user how to do anything they need to know about the system including how to fix it if an error were to occur. Any maintenance can be done through web server using content management software when needed.

Conversion

We decided to implement our prototype using the direct conversion style. We know that the risks are high but the low cost and it lowered the time required which was more important to us to meet the requirements we had.

User training

Training is designed to very quick and easy. All warehouse employees will be taught how to navigate the system and how to create new information or edit existing information. The downside to our system which was un-avoidable was that it takes quite a bit of time to enter in data into each field when creating a new product and it can be time consuming to resolve discrepancies of any come up. The resolve discrepancy function is there to help find and replace error more quickly then to go through each line of information to see what you did wrong. It shouldn't take more than thirty minutes to teach the employees how to use our system since it is extremely straightforward.

Intro to prototype

Our prototype is designed to show an easier way to create, edit and update product information to the database. Due to the nature of our product, as a web-based application, we have chosen to use the Extreme Prototyping method. While using this method we used html for the prototype so whatever web browser you prefer to use will be able to run our system. We designed the system to be easier to navigate between each page and to be able to add new information easily to the databases. The way we have it designed right now anyone that knows how to navigate a web site could easily learn to use our system which makes it helpful for the business that will use what we created. There would not be a limit on who knows how to use the system because it is so easy to use. This is helpful for the regular employees that need to edit or add new information within the system.

Functionality included

- Create New Product Information
- Edit Existing Product Information
- Check Product Information submitted for Discrepancies
- Resolving Discrepancies if there are any
- View Current Quantity-On-Hand and Inventory Information based on Warehouse location

Functionality excluded

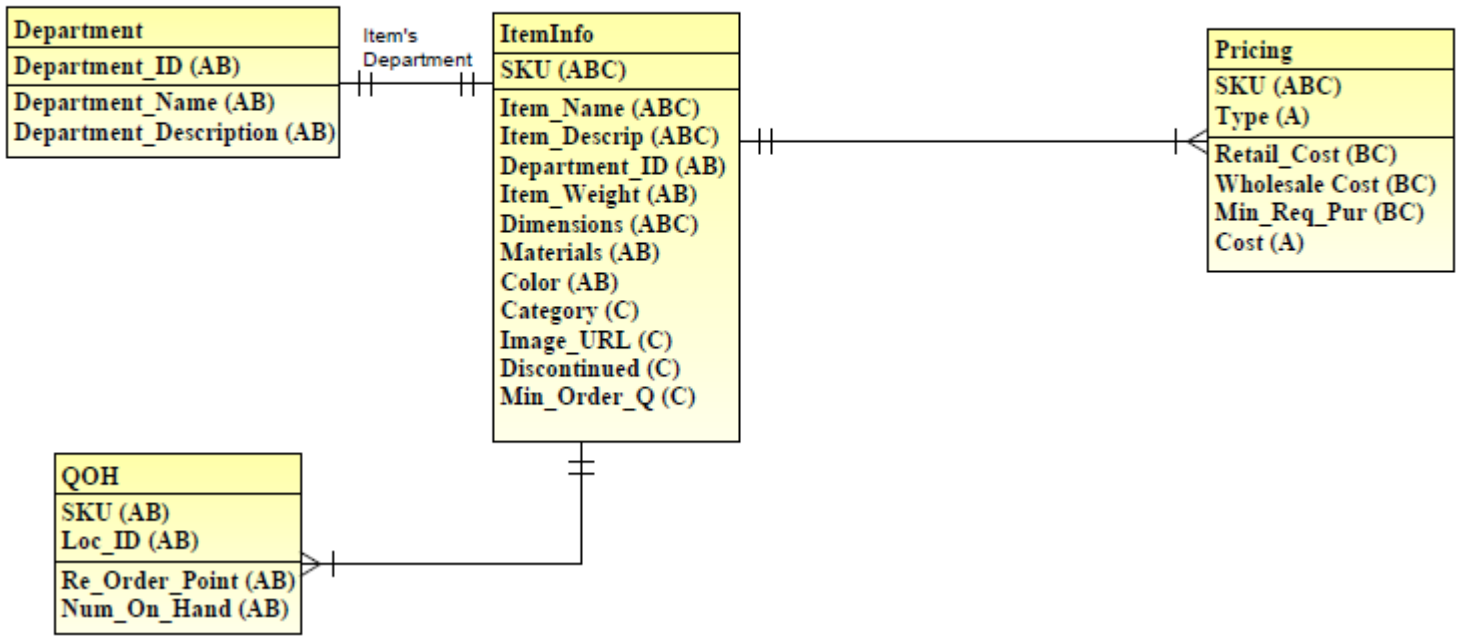
- Updating information changed to the database
- Integration with Web Services (Internet)

APPENDIX

Final Report Content (Appendix)

Relational Database Model (ERD – Visible Analyst Generated)	Page A1
Create Table Scripts (MySQL Generated)	Page A2
Scheme Screenshot (MySQL Generated)	Page A4
Test Scenarios	Page A5
Prototype Screenshots	Page A7
Sample Prototype Code	Page A13

Team Logical Partition - Final ERD



Create Schema – MySQL Database Code

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

CREATE SCHEMA IF NOT EXISTS `proof_of_concept` DEFAULT CHARACTER SET latin1 ;
USE `proof_of_concept` ;

-- -----------------------------------------------------
-- Table `proof_of_concept`.`department`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `proof_of_concept`.`department` (
  `Department_ID (AB)` INT(11) NOT NULL ,
  `Department_Name (AB)` VARCHAR(50) NULL DEFAULT NULL ,
  `Department_Descrip (AB)` VARCHAR(255) NULL DEFAULT NULL ,
  PRIMARY KEY (`Department_ID (AB)`) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;


-- -----------------------------------------------------
-- Table `proof_of_concept`.`iteminfo`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `proof_of_concept`.`iteminfo` (
  `SKU (ABC)` INT(11) NOT NULL ,
  `Item_Name (ABC)` VARCHAR(50) NOT NULL ,
  `Item_Descrip (ABC)` VARCHAR(255) NULL DEFAULT NULL ,
  `Dept_ID (AB)` INT(11) NULL DEFAULT NULL ,
  `Item_Weight (AB)` VARCHAR(45) NULL DEFAULT NULL ,
  `Dimensions (ABC)` VARCHAR(45) NULL DEFAULT NULL ,
  `Color (AB)` VARCHAR(45) NULL DEFAULT NULL ,
  `Category (C)` VARCHAR(45) NULL DEFAULT NULL ,
  `Image_URL (C)` TEXT NULL DEFAULT NULL ,
  `Discontinued (C)` TINYINT(4) NULL DEFAULT NULL ,
  `Min_Order_Q (C)` SMALLINT(6) NULL DEFAULT NULL ,
  `Materials (AB)` TINYTEXT NULL DEFAULT NULL ,
  PRIMARY KEY (`SKU (ABC)`) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
```

Create Schema – MySQL Database Code (Continued)

```
-- -----------------------------------------------------
-- Table `proof_of_concept`.`pricing`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `proof_of_concept`.`pricing` (
  `SKU (ABC)` INT(11) NOT NULL ,
  `Type (A)` VARCHAR(30) NOT NULL ,
  `Cost (A)` DECIMAL(10,0) NULL DEFAULT NULL ,
  `Retail_Cost (BC)` VARCHAR(45) NULL DEFAULT NULL ,
  `Wholesale_Cost (BC)` DECIMAL(10,0) NULL DEFAULT NULL ,
  `Min_Req_Pur (BC)` SMALLINT(6) NULL DEFAULT NULL ,
  PRIMARY KEY (`SKU (ABC)`, `Type (A)`) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;


-- -----------------------------------------------------
-- Table `proof_of_concept`.`qoh`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `proof_of_concept`.`qoh` (
  `SKU (AB)` INT(11) NOT NULL ,
  `ID (AB)` INT(11) NOT NULL ,
  `Re_Order_Point (AB)` SMALLINT(6) NULL DEFAULT NULL ,
  `Num_On_Hand (AB)` SMALLINT(6) NOT NULL ,
  PRIMARY KEY (`SKU (AB)`, `ID (AB)`) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;



SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Schema – MySQL Database

```
+----------------------------+
| Tables_in_proof_of_concept |
+----------------------------+
| department                 |
| iteminfo                   |
| pricing                    |
| qoh                        |
+----------------------------+
4 rows in set (0.02 sec)

mysql> describe department
    -> ;
+-------------------------+-------------+------+-----+---------+-------+
| Field                   | Type        | Null | Key | Default | Extra |
+-------------------------+-------------+------+-----+---------+-------+
| Department_ID (AB)      | int(11)     | NO   | PRI | NULL    |       |
| Department_Name (AB)    | varchar(50) | YES  |     | NULL    |       |
| Department_Descrip (AB) | varchar(255)| YES  |     | NULL    |       |
+-------------------------+-------------+------+-----+---------+-------+
3 rows in set (0.02 sec)

mysql> describe iteminfo;
+--------------------+-------------+------+-----+---------+-------+
| Field              | Type        | Null | Key | Default | Extra |
+--------------------+-------------+------+-----+---------+-------+
| SKU (ABC)          | int(11)     | NO   | PRI | NULL    |       |
| Item_Name (ABC)    | varchar(50) | NO   |     | NULL    |       |
| Item_Descrip (ABC) | varchar(255)| YES  |     | NULL    |       |
| Dept_ID (AB)       | int(11)     | YES  |     | NULL    |       |
| Item_Weight (AB)   | varchar(45) | YES  |     | NULL    |       |
| Dimensions (ABC)   | varchar(45) | YES  |     | NULL    |       |
| Color (AB)         | varchar(45) | YES  |     | NULL    |       |
| Category (C)       | varchar(45) | YES  |     | NULL    |       |
| Image_URL (C)      | text        | YES  |     | NULL    |       |
| Discontinued (C)   | tinyint(4)  | YES  |     | NULL    |       |
| Min_Order_Q (C)    | smallint(6) | YES  |     | NULL    |       |
| Materials (AB)     | tinytext    | YES  |     | NULL    |       |
+--------------------+-------------+------+-----+---------+-------+
12 rows in set (0.00 sec)
```

Schema – MySQL Database (Continued)

```
mysql> describe pricing;
+---------------------+--------------+------+-----+---------+-------+
| Field               | Type         | Null | Key | Default | Extra |
+---------------------+--------------+------+-----+---------+-------+
| SKU (ABC)           | int(11)      | NO   | PRI | NULL    |       |
| Type (A)            | varchar(30)  | NO   | PRI | NULL    |       |
| Cost (A)            | decimal(10,0)| YES  |     | NULL    |       |
| Retail_Cost (BC)    | varchar(45)  | YES  |     | NULL    |       |
| Wholesale_Cost (BC) | decimal(10,0)| YES  |     | NULL    |       |
| Min_Req_Pur (BC)    | smallint(6)  | YES  |     | NULL    |       |
+---------------------+--------------+------+-----+---------+-------+
6 rows in set (0.02 sec)

mysql> describe qoh;
+---------------------+-------------+------+-----+---------+-------+
| Field               | Type        | Null | Key | Default | Extra |
+---------------------+-------------+------+-----+---------+-------+
| SKU (AB)            | int(11)     | NO   | PRI | NULL    |       |
| ID (AB)             | int(11)     | NO   | PRI | NULL    |       |
| Re_Order_Point (AB) | smallint(6) | YES  |     | NULL    |       |
| Num_On_Hand (AB)    | smallint(6) | NO   |     | NULL    |       |
+---------------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Sample test scenario.

Test Scenario 1.

Program ID	Logical Partition	Version	1
Tester	Hugh Hunton	Date Designed	April 15, 2012
		Date Conducted	Should be left blank

Test ID:	1	Requirement Addressed:	Edit existing information

Objective	Ensure that the program correctly changes the fields I edit within the system
Procedure	Open product information form, select Edit button, enter new data as shown in test cases, press Submit button.

Test Cases

Test case Number	Interface ID	Data Field	Data Entered
1	Edit product information form	Item Name	“Futon”
		Retail Cost	800
		Weight	140
2	Edit product information form	Item Name	“Desk Chair”
		Sku	CH23
		Weight	15
2	Edit product information form	Item Name	“Coffee table”
		Weight	80
		Retail Cost	225

Expected Results	Case 1 is accepted, cases 2 and 3 are accepted results are added
Actual results	Should be left blank

Sample test scenario.

Test Scenario 1.

Program ID	Logical Partition	Version	1
Tester	Hugh Hunton	Date Designed	April 15, 2012
		Date Conducted	Should be left blank

Test ID:	2	Requirement Addressed:	Create new product

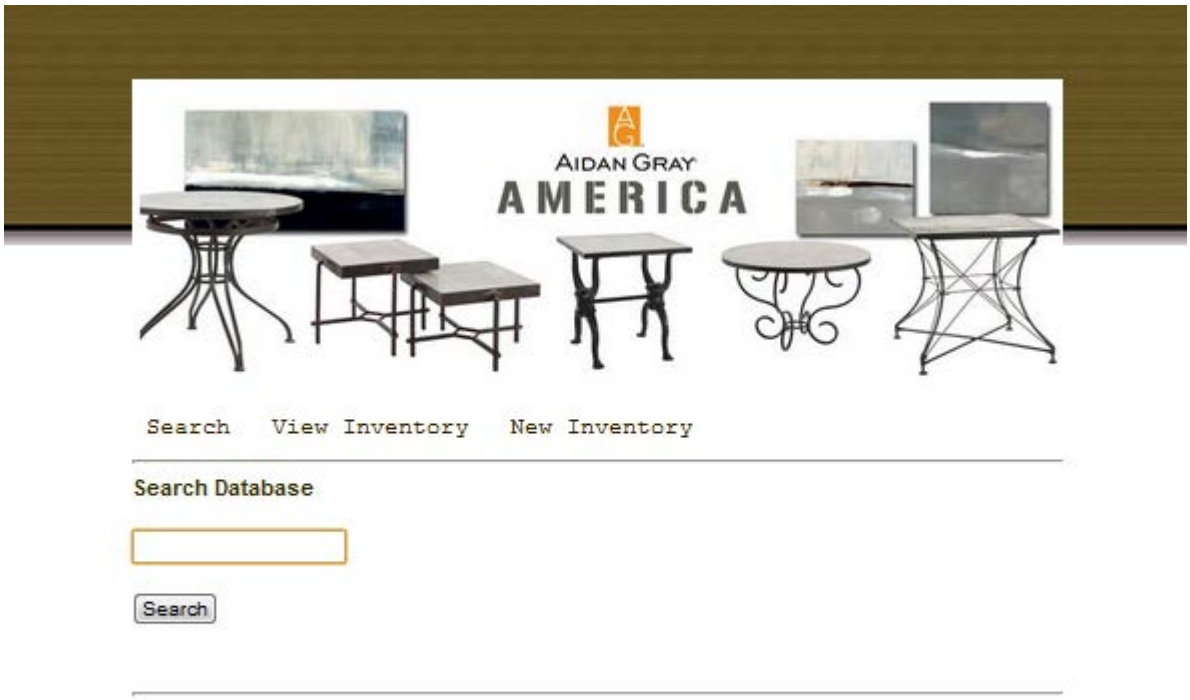
Objective	Ensure that the program correctly creates a new product
Procedure	Open new product form, enter new data as shown in test cases, press Submit button.

Test Cases

Test case Number	Interface ID	Data Field	Data Entered
1	New Product form	Type	“Garden”
		Color	“Green”
		Width	120
2	New Product form	Wholesale cost	200
		Sku	KEH1245
		Weight	50
2	New Product form	Minimum Quantity	6
		Department ID	9405
		Material	“Copper”

Expected Results	Case 1 is accepted, cases 2 and 3 are accepted results are added
Actual results	Should be left blank

Prototype – The Welcome Page



Form Page - Product Information (Uneditable)

Search View Inventory New Inventory

Product Information

Edit

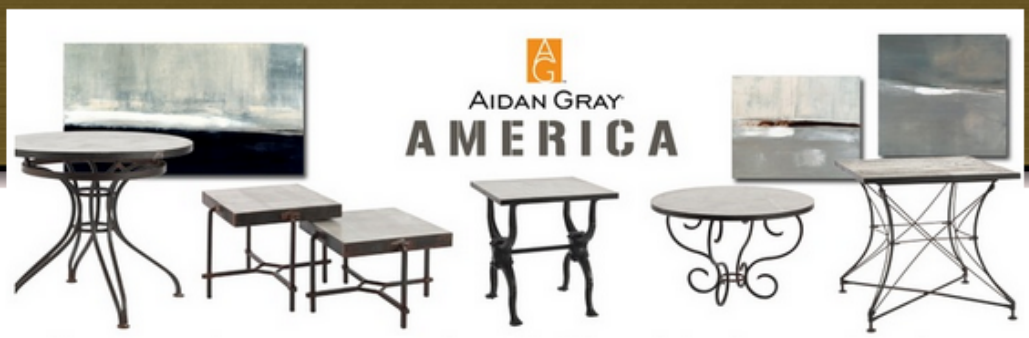
Sku:
CH22

Item Name:
Ethan Dining Chair

Item Description:
Constructed from European Oak each frame is carved and then burnished to a rustic patina. The chair is finished with Light Linen and Burlap accents on back, rusted nails hold the fabric in place. Exposed nail heads lend to a more uneven edge appearance and bumps, unevenness and roughness should be expected.

Image URL:
image.aidengray.org

Form Page – Product Information (Editable)



Search View Inventory New Inventory

Product Information

Sku:
CH22
Item Name:
Ethan Dining Chair
Item Description:
Constructed from European Oak each frame is carved and then burnished to a rustic patina. The chair is finished with Light Linen and Burlap accents on back, rusted nails hold the fabric in place. Exposed nail heads lend to a more uneven edge appearance and bumps, unevenness and roughness should be expected.
Image URL:
image.aidengray.org
Type:
Furniture

Form Page – Successful Edit

Search View Inventory New Inventory

Your product edit was successful!

Form Page – New Product Added

Search View Inventory New Inventory

Your new product was successfully added!

Product Information

Sku:
KEH1234
Item Name:
Decorative Lawn Gnor
Item Description:
A beautiful lawn gnome that will inspire jealousy in your neighbors
Image URL:
gnomey.jpg
Type:
Garden
Retail Cost:
$57.99
Wholesale Cost:
$47.99
Cost:
$37.99
Minimum Quantity:
2

Report Page – Current Inventory Report

Search View Inventory New Inventory

Current Inventory

Item Number	Item Name	Location Number	Quantity
CH22	Armoire	1	27
CH22	Armoire	2	0
CH22	Armoire	3	4
OL15	Leather Chair	1	9
OL15	Leather Chair	2	13
OL15	Leather Chair	3	0
FN17	Vinyl Couch	1	5
FN17	Vinyl Couch	2	32
FN17	Vinyl Couch	3	18
ZZ12	Lawn Chair	1	12
ZZ12	Lawn Chair	2	15
ZZ12	Lawn Chair	3	35
CH22	Wooden Desk	1	27
CH22	Wooden Desk	2	13
CH22	Wooden Desk	3	5
OF17	Office Chair	1	12
OF17	Office Chair	2	19
OF17	Office Chair	3	14

HTML Code – Search.HTML (Home Page)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta name="keywords" content="" />
<meta name="description" content="" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Logical Partition</title>
<link href="style.css" rel="stylesheet" type="text/css" media="screen" />
</head>
<body>
<!-- end #header -->
<!-- end #menu -->
<div id="page">
                <div id="page-bgtop">
                        <div id="page-bgbtm">
                                <div id="content">
                                        <div class="post">
<p><img src="images/img04.jpg" width="620" height="200" alt="" /></p>

                <div class="auto-style1"><p>
<PRE> <A HREF="SearchClean.html">Search</A>  <A HREF="Inventory.html">View Inventory</A>  <a
href="New.html">New Inventory</a></PRE></p>
                </div></form><hr />
<b>Search Database</b> <br /><br />
<form name="Main" action="SearchFilled.html" method="post">
<INPUT type="text" value="Enter Sku (CH22)" onfocus="if
(this.value==this.defaultValue) this.value='';">
<br />
<br>

<INPUT TYPE="submit" Name="redirect" VALUE="Search">
</FORM>
<br />
<br />
<hr />
</body>
</html>
```

HTML Code – New.HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta name="keywords" content="" />
<meta name="description" content="" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Logical Partition</title>
<link href="style.css" rel="stylesheet" type="text/css" media="screen" />
</head>
<body>
<!-- end #header -->
<!-- end #menu -->
<div id="page">
                    <div id="page-bgtop">
                                        <div id="page-bgbtm">
                                                            <div id="content">
                                                                                <div class="post">
                    <p><img src="images/img04.jpg" width="620" height="200" alt="" /></p>
<p>
      <PRE> <A HREF="SearchClean.html">Search</A>   <A HREF="Inventory.html">View Inventory</A>
<a href="New.html">New Inventory</a></PRE>
                                        </p> </form><hr />
<b>New Product</b> <br /><br />
<br />
<b>Enter Product Information</b> <br /><br />
Sku: <br/><INPUT type="text" value="(KEH1234)" onfocus="if
(this.value==this.defaultValue) this.value='';">
<br />
Item Name: <br /> <INPUT type="text" value="(Decorative Lawn Gnome)" onfocus="if
(this.value==this.defaultValue) this.value='';">
<br />
Item Description:<br/> <INPUT type="text" value="(Beautiful Lawn Gnome that inspires jealous in your
neighbors)" onfocus="if
(this.value==this.defaultValue) this.value='';">
<br />
Image URL: <br/> <INPUT type="text" value="(gnomey.jpg)" onfocus="if
(this.value==this.defaultValue) this.value='';"> <br />
```

HTML Code – New.HTML (Continued)

```
Type:<br/>
<select name="category">
                    <option value="lighting">Lighting</option>
                    <option value="decor">Furniture</option>
                    <option value="furniture" SELECTED>Furniture</option>
                    <option value="candle">Candle</option>
                    <option value="garden">Garden</option>
</select><br/>
Retail Cost: <br/><INPUT type="text" value="(57.99)" onfocus="if
(this.value==this.defaultValue) this.value='';"> <br />
Wholesale Cost: <br/> <INPUT type="text" value="(47.99)" onfocus="if
(this.value==this.defaultValue) this.value='';"> <br />
Cost: <br/> <INPUT type="text" value="(37.99)" onfocus="if
(this.value==this.defaultValue) this.value='';"> <br />
Minimum Quantity:<br/>
<select name="minimumQuantity">                    <br />
                    <option value="0">0</option>
                    <option value="2">2</option>
                    <option value="4">4</option>
                    <option value="6">6</option>
</select><br/>
<br /><br />
<b> Department Information </b>
<Br /><br />
Department ID: <br/> <input type="text" name="price" value=""></input><br/>
Department Name: <br/> <input type="text" name="price" value=""></input><br/>
Department Description: <br/> <input type="text" name="price" value=""></input><br/><br />
<b> Quantity Information </b> <br /><br />
ID: <br/> <input type="text" name="price" value=""></input><br/>
Reorder Point: <br/> <input type="text" name="price" value=""></input><br/>
Color: <br/> <input type="text" name="color" value=""></input><br>
Weight:<br/> <input type="text" name="width" size="7" value="">
</input><br/>
Height (inches):<br/> <input type="text" name="height" size="7" value=""></input><br/>
Width (inches):<br/> <input type="text" name="width" size="7" value=""></input><br/>
Depth (inches):<br/> <input type="text" name="depth" size="7" value=""></input><br/>
Material 1:<br/> <input type="text" name="material1" value=""></input><br/>
Material 2:<br/><input type="text" name="material2" value=""></input><br/>
```

HTML Code – New.HTML (Continued)

```
Material 2:<br/><input type="text" name="material2" value=""></input><br/>
Case Dimensions (Cubic Meters):<br/> <input type="text" name="cbm" value=""></input><br/>
        <form name="Main" action="NewFilled.html" method="post">
        <INPUT TYPE="submit" Name="redirect" VALUE="Submit">
                                                    </FORM>
</body>
</html>
```

HTML Code – Inventory.HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta name="keywords" content="" />
<meta name="description" content="" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Logical Partition</title>
<link href="style.css" rel="stylesheet" type="text/css" media="screen" />
</head>
<body>
<!-- end #header -->

<!-- end #menu -->
<div id="page">
                <div id="page-bgtop">
                                <div id="page-bgbtm">
                                                <div id="content">
                                                                <div class="post">
                                <p><img src="images/img04.jpg" width="620" height="200" alt=""
/></p>
                <div class="auto-style1">
                                <p>
<PRE> <A HREF="SearchClean.html">Search</A>  <A HREF="Inventory.html">View Inventory</A>  <a
href="New.html">New Inventory</a></PRE>
                                </p>

                </div>
  </form>
                <hr />
<b>Current Inventory</b> <br /><br />

<br />
<br />
</body>
</html>
```